

Wallenium's Plugin Class Framework

Developer Guide

Last changed: 14.11.2007
© Simon Wallmann

Initialize the Plugin Framework

To use the Plugin Framework in all files, create a new file named 'common.php' and include it in every file you want to use the plugin framework. I added the table-defines in that file, too, you save much time and size if you do it once instead of in every file header.

```
include_once($eqdkp_root_path . 'plugins/raidplan/includes/wpfc/init.pwc.php');  
$wpfccr = new InitWPFC($eqdkp_root_path.'plugins/raidplan/includes/wpfc/');
```

this is the Code, if the Plugin Framework (in this example raidplan) is in the /include/wpfc/ folder. You might not change that, its easier ☺

Database Upgrade Warning Messages

If you're using Database tables of other plugins and you want to check, if all plugins are up to date. You need: a saved plugin version in the database

```
$pluginlist      = array('raidplan', 'charmanager');  
$pluginconfigs  = array(  
    'raidplan'    => array(  
        'table' => 'raidplan_config',  
        'field' => 'rp_inst_version'  
    ),  
    'charmanager' => array(  
        'table' => 'charmanager_config',  
        'field' => 'uc_inst_version'  
    )  
);
```

Its pretty easy: \$pluginlist contains all plugins to check. \$pluginconfig contains the database table and the field in that table which contains the version of the plugins.

After that you can use the plugin warning, and place it in the Template:

```
'UPGRADEWARN' => $wpfccore->InitUpgradeWarn($pluginlist, $pluginconfigs),
```

Add the following in the template header

```
{UPGRADEWARN}
```

That's all. If the plugins are not up to date, a warning box will be displayed. You can avoid errors by using tables which does not exist.